# Reality Gates: Un-Foolable Exact Certificates for AI-Proposed Scientific Hypotheses

**Alexander Vicol**

University of Toronto

## Abstract

Large language models propose scientific hypotheses fluently, but a user cannot tell which confident outputs to trust. We argue that the fix is architectural: separate an untrusted *proposer* (any model) from a *verifier* that is an *exact reality gate*, a decision procedure grounded in a conservation law, algebraic identity, or rigorous inequality whose verdict is a machine-checkable theorem rather than a judgment, and enforce an *honest-tiering* invariant, proved sound: only an exact gate may mint the label VERIFIED, and no composition of a fluent proposer, web search, code execution, or device measurement can raise the tier. Our main theoretical result makes this necessary, not merely safe. We prove that any trust method reading only the proposer's own output (confidence, self-consistency, self-critique, LLM-as-judge) is ROC-dominated by the proposer and cannot reach the ideal operating point unless the proposer already decides the claim; we measure that it does not, since across all four frontier models the proposer's own signal separates true from false hard claims at only $\mathrm{AUC} \in [0.66, 0.87]$, and at chance on the compute-hard ones. External exact grounding is therefore *necessary and sufficient* to certify, and the empirical observation that no LLM certifies becomes an impossibility rather than a capability gap. The matching utility result is cast as selective prediction (Fig. 1): on a 4-domain, exact-labeled benchmark ($n$=240), the exact gate is the only method at the ideal corner $(\text{coverage } 1,\ \text{false-accept } 0)$ and strictly Pareto-dominates every frontier LLM and every LLM-based trust method, including four *tool-augmented* models. A code interpreter relocates the failure (to abstention, residual false-accept, or over-rejection) but never removes it, exactly as the theorem predicts, since a tool is a step toward external grounding. A weaker model's dangerous false-accept rate rises $16\times$ $(0.017 \to 0.271)$ with problem size, and self-critique and self-consistency do not close the gap. Nor is the gate merely a filter: the same trust floor makes open-ended autonomy both safe and productive. Verifier-guided, *actionable* exact feedback with a gate-supervised policy lifts an unreliable proposer's held-out solve rate from $0.00$ to $0.92$; run open-loop under the gate, the same proposer keeps only verified outputs, and they are non-trivial: on the Erdős–Straus hard core it returns machine-checkable decompositions such as $4/1009 = 1/253+1/85100+1/944524900$ at zero false-accept. The discipline spans number theory, particle physics, quantum many-body systems, logic circuits, and finance; it adjudicates $27/27$ cross-domain debates (against $26/39$ for a configuration-trusting baseline) and never mints a false certificate across nine real IBM quantum jobs. Finally, we explain why the gate is needed: efficiency indicates truth because truth is *coherence*, so efficiency *proposes* and only an exact disharmony gate *disposes*. Ungated, efficiency is fooled on $100\%$ of constructed cases where a false hypothesis is simpler than the true one, the description-length face of reward hacking; the gate is fooled on none. The deliverable is a trust layer that is Pareto-optimal for selective prediction and cannot be fooled: safe and useful by construction.

## 1 Introduction

Asked whether $\mathtt{cccc\bar{c}}$ is a legitimate hadron, whether $1/1901 + 1/16629 + 1/19098 = 10709261/16778843919$, or for a tetraquark's mass, a capable language model answers fluently, confidently, and sometimes wrongly. This is the central obstacle to using generative models as scientific collaborators: their outputs carry no reliable internal signal separating the theorem-grade from the plausible-but-false. For science an *unmarked* error rate is often worse than useless, because the errors are exactly the novel claims one would most want to trust.

We take the position that trust should not be sought inside the proposer at all. We *factor* the system into a fast, creative **proposer** (any model) and a slow, narrow **verifier** that is an *exact reality gate*: it does not *estimate* whether a hadron is colour-neutral, it *computes* $(n_q - n_{\bar{q}}) \bmod 3$ in exact integer arithmetic and returns REFUTED if the result is nonzero. It cannot be prompt-engineered and does not soften when the proposer is more persuasive. Around this factoring we enforce one invariant, **honest tiering**: only an exact gate may mint the trust label VERIFIED; a proposer, web search, fitted model, or noisy device is tiered strictly below, as a *lead* or an *abstention*, and no composition raises the tier (Section 3, Theorem 1).

**Contributions.**

1. **Two theorems: certification is impossible internally, and sound only through an exact gate** (Section 3). (a) A *no-internal-certifier* theorem shows that any trust method reading only the proposer's own output is ROC-dominated and cannot reach the ideal $(\text{coverage } 1,\ \text{false-accept } 0)$ unless the proposer already decides the claim, so external grounding is *necessary and sufficient*. (b) An honest-tiering soundness theorem shows that only an exact gate mints VERIFIED, and that the label cannot be forged under any composition of proposer, web,

code, or device. Together, nothing weaker than an external exact gate can certify, and the gate cannot be fooled. Governance (what the system may do) is orthogonal to certification (what it may claim).

2. **A false-claim benchmark that evaluates the trust layer as an AI method** (Section 5): an exact-labeled corpus across four domains and two difficulty tiers, with frontier baselines, standard LLM trust methods, and a difficulty axis, *plus* a natural corpus of claims a model spontaneously emits (gate-labeled). The headline is a negative about LLMs and a positive about gates: no LLM method certifies, not even on its own organic output.
3. **A verifier-guided-generation method** (Section 6): the gate makes an unreliable proposer sound; and while a *scalar* error signal does not improve the proposer, *actionable* exact-sub-result feedback plus a gate-supervised learned proposal policy lift its solve rate from $0.00$ to $0.92$ on held-out problems. The feedback *design*, not the loop, was the barrier. The recipe generalizes across three structurally different exact domains (Erdős–Straus, Fermat factorization, sum of two squares): the actionable-feedback lift is universal; the learned-prior lift is transfer-gated, and we characterize that boundary.
4. **Cross-domain gate instances** (Section 4), including an adversarial quantum-hardware stress-test in which the invariant refuses to certify a noisy result across nine real jobs and the full mitigation stack, and a debate-adjudication instance in which the gate *chooses between* competing hypotheses ($27/27$ vs. a $26/39$ configuration-trusting baseline).
5. **A mechanism for why the gate is necessary** (Section 4): efficiency indicates truth because truth is coherence, and coherence compresses. Naked efficiency, however, is fooled precisely by false-but-simple hypotheses (reward hacking), which we construct explicitly: min-description-length selection is fooled on $100\%$ of the adversarial cases, and an exact disharmony gate on $0\%$.

All components ship as runnable experiments with passing exact-gate tests; every number is read from a committed artifact and every LLM verdict is cached and reproducible at zero cost.

**A new kind of co-discovery system, and its efficiency.** Taken together the components describe not a filter bolted onto a chatbot but a distinct class of *co-discovery* system: an untrusted generative proposer bound to an exact verifier by a provable honesty invariant, so the pair is creative like a model yet sound like a proof kernel, and stays honest even when it acts on the open web or tries to improve itself (Section 7). Two efficiency properties make this practical, not merely safe. *Efficiency of trust*: the certifying step is typically far cheaper than the claim (an integer $\bmod 3$ instead of a mass computation, a variational *box* instead of a diagonalization), so certification does not inherit the cost of discovery. *Efficiency of search*: because the gate returns exact structure rather than a scalar, a cheap proposer under gate-supervised feedback reaches held-out solutions in $\sim 2$ rounds (Section 6), so the expensive frontier model is not needed for the trustworthy part. Trustworthiness is thus a *safe-by-design* architectural property with a low, bounded price (Theorem 1), not a capability bought with scale.

## 2 Related Work

**Verifier-guided generation and verifiable rewards.** Filtering model output through an external checker is established in program synthesis and theorem proving, where proof kernels drive retrieval-augmented provers and olympiad solvers (Yang et al. 2023; Trinh et al. 2024); reinforcement learning from *verifiable* rewards (DeepSeek-AI 2025), together with process and outcome reward models (Lightman et al. 2023), likewise trains models against exact checkers. Those methods use the checker as a *training signal* or a soft re-ranker; our contribution is orthogonal and sits at inference time. We import into agentic LLM certification the programming-languages notion of a *soundness invariant*, familiar from type soundness and proof-carrying code (Necula 1997), where a machine-checkable certificate lets a consumer trust an artifact without trusting its producer. Here the checker's *tier* propagates as a hard, non-forgeable label that is sound *by construction* (Theorem 1) rather than by empirical calibration, and it cannot be forged by any composition of proposer, web search, code execution, or device measurement. What that buys is what we evaluate (a false-claim benchmark; the soundness theorem; a learned verifier-guided proposer; an adversarial honesty stress-test), not a new checker and not a solver.

**LLM reliability.** Self-consistency (Wang et al. 2023), self-refinement (Madaan et al. 2023), and LLM-as-judge (Zheng et al. 2023) improve accuracy but remain *internal* to the model's distribution; an LLM checking an LLM shares its blind spots, and hallucination persists (Ji et al. 2023). A parallel line *detects* likely hallucinations from sampling inconsistency (Manakul, Liusie, and Gales 2023) or semantic-entropy uncertainty (Farquhar et al. 2024). This is valuable, but it *estimates* a probability of error rather than *certifying* its absence: the output is a calibrated score, not a machine-checkable theorem, so it neither reaches $(0, 0)$ nor composes into a non-forgeable trust label. Conformal methods add a *statistical* guarantee: conformal factuality (Mohri and Hashimoto 2024) and conformal risk control (Angelopoulos et al. 2021) calibrate an answer/abstention set so a target error rate $\alpha$ holds with high probability over an exchangeable calibration sample. This is a genuine guarantee, but a distributional one: it bounds error at a chosen $\alpha{>}0$ rather than certifying its absence, still admits false-accepts at that rate, and can degrade under distribution shift, because it constrains a population rather than the individual output. Our gate is the deductive, $\alpha{=}0$ end of that spectrum: narrower in coverage, but sound *by construction* where it applies, independent of any calibration distribution and unaffected by shift. We measure the contrast directly (Section 5): self-critique and self-consistency do not reach zero error, and an external exact gate does.

**AI for science and quantum error mitigation.** Learned proposers make discoveries in mathematics and the sciences (Romera-Paredes et al. 2024); separately, zero-noise extrapolation, dynamical decoupling, and twirling mitigate

NISQ noise (Temme, Bravyi, and Gambetta 2017; Kandala et al. 2019). We connect them: the true NISQ bottleneck is *trusting* a noisy output, and an exact conservation-law gate is a certificate for it (Section 4).

# 3 The Reality-Gate Method

## 3.1 Propose → gate → dispose

A task is handled by a proposer emitting candidate objects (a hadron content, a decomposition, a circuit, an amplitude) and a bank of exact gates each returning one of {VERIFIED, REFUTED, ABSTAIN} with a certificate. The proposer only *narrows* an intractable space; the gate decides, and is the sole source of the VERIFIED tier.

## 3.2 The honest-tiering invariant, made sound

Tiers are totally ordered, HEURISTIC < REFERENCE < EMPIRICAL < VERIFIED. A tool result is *proof-bearing* iff its status is in {VERIFIED, REFUTED} *and* its tier is VERIFIED, which holds iff it is the output of an exact gate $g$ in the gate set $G$. Three implementation guards, each independently sufficient, make the invariant non-forgeable (proofs in the artifact): *(L1, per-tool discipline)* every non-gate tool returns a status/tier that is not proof-bearing; *(L2, whole-run clamp)* an autonomous multi-tool answer is capped at HEURISTIC; *(L3, product map)* the only aggregation edge yielding VERIFIED is the gate map `from_gate`.

**Theorem 1** (Soundness of certification)**.** *For any query and run, if the answer $a$ is badged VERIFIED then some result $r_i$ is proof-bearing; that is, $a$ is exactly the certified statement of an exact gate $g \in G$. Contrapositive: if no exact gate fired, $a$ cannot be badged VERIFIED.*

*Proof sketch.* If $\tau(a) =$ VERIFIED, by L3 the only edge producing it is `from_gate` on a result with status in {VERIFIED, REFUTED} from a gate, which is proof-bearing. A synthesized cross-tool answer is capped by L2 at HEURISTIC < VERIFIED; and by L1 no non-gate result is proof-bearing, so `from_gate` cannot have fired on one. Hence some $r_i$ is proof-bearing. □

Corollaries: the tier is un-foolable under tool composition and under proposer pressure, and absent a firing gate the maximal attainable tier is EMPIRICAL (a lead), so on open problems the system abstains rather than certifies.

**Why the gate is *necessary*, not merely sound.** Theorem 1 says the gate is safe; the next result says nothing weaker will do. For an exactly-decidable claim family $C$ with gold predicate $g(c) \in \{\mathrm{T}, \mathrm{F}\}$, call a trust method $M$ *proposer-internal* if its verdict $\in \{\mathsf{accept}, \mathsf{reject}, \mathsf{abstain}\}$ is a (possibly randomized) function of a fixed proposer $P$'s own output on $c$ and invokes no external decision procedure for $g$; confidence thresholding, self-consistency, self-critique, and LLM-as-judge are all internal. Let $\alpha = \Pr[\mathsf{accept} \mid g{=}\mathrm{T}]$, $\beta = \Pr[\mathsf{accept} \mid g{=}\mathrm{F}]$.

**Theorem 2** (No internal certifier)**.** *Every proposer-internal $M$ has operating point $(\beta, \alpha)$ on or under the ROC of $P$'s best statistic on $C$; hence $\alpha - \beta \le 2\,\mathrm{AUC}_P - 1$. The ideal certifier point $(\alpha, \beta) = (1, 0)$ is internally attainable* iff $\mathrm{AUC}_P = 1$. *Thus on any sub-family with $\mathrm{AUC}_P < 1$ no internal method certifies, while an external exact gate, which computes $g$, attains $(1, 0)$. External grounding is* necessary and sufficient.

(Neyman–Pearson plus data processing; full proof in the appendix.) This turns the benchmark's central negative into an *impossibility*: LLM trust methods miss the ideal corner not for want of scale but because a signal that cannot decide $g$ cannot certify $g$. It also unifies the tool result of the Discussion: a code tool helps precisely because it moves *toward* an external decision, raising the effective AUC where a claim is codeable, and the reality gate is that external decision, at the corner by construction. We measure $\mathrm{AUC}_P$ on the hard family next.

# 4 Exact Gate Instances

The trust discipline is not tied to any one domain: a reality gate exists wherever an exact law does, so the same VERIFIED/REFUTED/ABSTAIN machinery spans number theory, particle physics, quantum many-body systems, and *software* (Table 1), each verdict a machine-checkable theorem rather than a fit. One instance doubles as an adversarial test of the honesty invariant itself. To reach an ordinary engineering domain we add a *logic-circuit* gate: structural rewrites propose a smaller netlist and a Z3 miter certifies exact equivalence to the specification, so a full adder cut $10 \to 3$ gates ($-62.5\%$ area) is VERIFIED and a non-equivalent rewrite is REFUTED (`chip_optimizer.json`). A second non-formal instance sharpens the *tiering*: a moving-average *backtest* whose three exact gates prove the execution sound (equity reconstructed bit-for-bit over 859 trades, 0 look-ahead leaks, exposure ≤ limit) while the *return* stays EMPIRICAL; the gate certifies the books, not the future (`exp_finance_gated_backtest.py`).

**Adversarial hardware honesty.** As the sharpest stress-test, we tried to force a noisy lattice-Schwinger (Kogut and Susskind 1975; Martinez et al. 2016) charge measurement on `ibm_marrakesh` ($N{=}3$) to *pass* the Gauss-law gate, using the full professional error-mitigation toolkit (readout mitigation, local and global zero-noise extrapolation, dynamical decoupling, twirling). Across nine real jobs the invariant returned a trustworthy REFUTED every time, and once *rejected* an available green "certified" number because an amplified point was unphysical; only a genuine physics fix (a free-fermion compilation cutting entangling gates $10 \to 4$) earned a pass. We claim no quantum advantage ($N{=}3$ is classically simulable); the invariant could not be fooled and rewarded a real fix (full narrative in the appendix).

**Adjudicating competing hypotheses.** The discipline extends from checking one claim to *choosing between* competing ones. Cast as a debate (two views, one shared measurement, an exact verifier that rejects the self-contradictory side), a gate-decided adjudicator is correct on all $27/27$ decidable cross-domain debates (number theory, chemistry, physics) and catches two deliberately mislabeled seeds, while a configuration-trusting baseline scores $26/39$ and flatlines exactly where its label is wrong (`results/exp_debate_multiverse_sweep`). A cheap

| domain | exact gate | REFUTEDS |
|---|---|---|
| number theory | divisor/residue identity, exact $\mathbb{Q}$ arithmetic | a wrong Egyptian-fraction identity |
| hadrons | colour-singlet triality $(n_q - n_{\bar{q}}) \bmod 3$ | a colour-impossible content |
| amplitudes | Ward identity: $\varepsilon \to k$ must vanish | a gauge-violating amplitude |
| many-body | Temple sandwich from $E$, $\langle H^2 \rangle$ | a ground-energy claim outside the box |
| hardware | Gauss law $[H, Q]{=}0$ (charge conserved) | a device run that broke conservation |
| logic circuits | Z3 miter: rewritten netlist $\equiv$ spec (SAT) | a non-equivalent circuit rewrite |
| finance/ backtest | equity reconstruction + no look-ahead + exposure | an accounting-drift or look-ahead leak |

Table 1: A cross-domain bank of exact gates; each verdict is a machine-checkable theorem, not a fit. The amplitude gate uses the Parke–Taylor structure (Parke and Taylor 1986); the many-body gate uses the Temple inequality (Temple 1928); the hardware gate certifies an exotic-hadron and Schwinger setting against LHCb (LHCb Collaboration 2022).

efficiency signal may *rank* the views, as in description-length model selection, but it is tiered strictly below the exact verdict and never mints VERIFIED. Efficiency proposes; the exact gate disposes.

**Why efficiency proposes, and why it must be gated.** This principle has a mechanism that is also the method's safety case against reward hacking. Efficiency indicates truth because truth is *coherence*: a true claim *composes* with the grounded record (cheap to extend, short to describe), while a false claim injects a contradiction that wastes search. The bridge is that coherence implies compressibility, which on exact batteries separates true from false at AUC 1.0 (Appendix). Efficiency *alone*, however, is foolable wherever a false hypothesis is *simpler* than the true one, the description-length face of a reward-hacking shortcut. We construct that adversary exactly: for a non-linear integer law, the line through its first two points is strictly simpler and matches on the seen window yet contradicts held-out grounded anchors. Over 60 such cases (43 adversarial), naked min-description-length selection is fooled on 100% of the adversarial cases, while gating the anchors first with an exact disharmony check, then ranking by efficiency, is correct on all 60 (Appendix). This is the general form of the finance instance above: the "just borrow more" policy is a false-but-simple, higher-reward shortcut the exposure gate rejects.

## 5 Evaluation I: A Benchmark for the Trust Layer

**Setup.** We build a balanced, exact-labeled corpus across four domains: Egyptian unit-fraction identities, modular arithmetic (primality, modular exponentiation, gcd), hadron quantum numbers (colour-singlet, charge, baryon number), and small-Hamiltonian ground-state energy. The generator produces up to 60 items/domain (a balanced $n{=}240$); we report the four-model comparison on a balanced 80-item slice per tier (20/domain, half valid, so all models share an identical corpus), and additionally validate the cheap model at the full $n{=}240$ and across three seeds. The exact gate *defines* truth (a test asserts gold=gate on every item). Invalid items are *genuinely hard*: false in ways that require exact computation to detect (an identity off by $1/930$; a semiprime claimed prime), not a surface cue. A *hard tier* scales numbers past the LLM-reliable range (e.g. $1/1901{+}1/16629{+}1/19098 \stackrel{?}{=} 10709261/16778843919$). The load-bearing metric is the **false-accept rate** (admitting a false claim). Baselines: constant predictors; LLM *direct*; LLM *self-critique*; LLM *self-consistency* (majority of $k{=}3$). Models: Gemini 2.5 Flash, Gemini 2.5 Pro, Claude Sonnet, and GPT-5.6 Sol (the four-model comparison; Gemini Pro's hard tier is $n{=}80$ by quota, marked in Table 2). All verdicts are cached.

**Result 1: no LLM certifies.** On the normal tier ($n{=}240$; Wilson 95% CIs) the gate is the only method at $(0, 0)$ with full coverage. `always-INVALID` also has zero false-accept, but at 100% false-reject; un-foolable certification needs both, and no LLM achieves both. Self-critique and self-consistency improve accuracy by $\leq 1.3$ points and do not close the gap. An LLM cannot certify claims it cannot compute (energy false-reject $\approx 1.0$).

**Result 2: at $n{=}240$, every model provably admits hard false claims.** Table 2 shows the difficulty curve, the central quantitative result. A weaker model's false-accept rises $\mathbf{16}\times$ on the hard tier ($0.017 \to 0.271$ at $n{=}240$; admitting $\mathbf{75}\%$ of false large-number identities, which it cannot verify with exact arithmetic), and a second, mid-frontier model (Gemini Pro) is worse still. Crucially, the larger sample overturns any appearance of frontier immunity: at $n{=}240$ *all four* models have a false-accept whose 95% CI *excludes zero*, including Claude Sonnet (0.010, driven by 3% of hadron claims) and GPT-5.6 Sol (0.075, admitting $\mathbf{30}\%$ of hard modular claims). Every LLM also fails the compute-hard ground-energy domain entirely. The effect replicates across three seeds (flash $0.213{\pm}0.033$). No LLM reaches $(0, 0)$; the gate is flat at $(0, 0)$ at both tiers, independent of difficulty, model, and seed.

**Result 3: the danger appears on organic model output, not just constructed items.** A fair objection to Table 2 is that we *designed* the false items. So we built a *natural* corpus: we asked GPT-4o-mini real exact questions (a three-term unit-fraction sum; a prime in a range; a quark content's charge) and gate-labeled its *own* answers. Of 86 spontaneously emitted claims the gate finds $\mathbf{23}\%$ *false* ($\mathbf{63}\%$ in fraction arithmetic, where a weak model genuinely mis-adds; 0% on prime-naming). These are organic errors the model asserted believing them true. When the *same model* verifies its own claims it admits $\mathbf{25}\%$ of the false ones (false-accept 0.250, 95% CI $[0.11, 0.47]$ excluding 0), while the exact gate is $(0, 0)$ at full coverage (Table 3). A stronger cross-verifier (Gemini 2.5 Flash) catches the slips but only by *abstaining*

| method (hard tier) | false-acc. [95% CI] | worst domain | acc. |
|---|---|---|---|
| EXACT GATE | **0.000** [0,.031] | — | 1.000 |
| Gemini 2.5 Flash | **0.271** [.196,.362] | Egyp. 0.75 | 0.664 |
| Gemini 2.5 Pro† | **0.256** [.146,.411] | Egyp/mod 0.40 | 0.671 |
| Claude Sonnet | **0.010** [.002,.055] | hadron 0.033 | 0.834 |
| **GPT-5.6 Sol** | **0.075** [.040,.136] | mod. 0.30 | 0.842 |

Table 2: Difficulty curve (hard tier, $n$=240; †Pro-hard $n$=80, Gemini quota). At $n$=240 *all four* models have a false-accept CI excluding zero: Flash and Pro admit roughly a quarter of hard false claims, and even Sonnet (0.010) and GPT-5.6 Sol (0.075) are provably nonzero. All LLMs fail the ground-energy domain (false-reject $\approx 1.0$). The gate is invariant to difficulty at $(0, 0)$.

| 86 organic (model-emitted) claims | false-acc. [95% CI] | false-rej. | acc. | cov. |
|---|---|---|---|---|
| **EXACT GATE** | **0.000** [0,.161] | **0.000** | 1.000 | **1.000** |
| GPT-4o-mini · self-verify | **0.250** [.112,.469] | 0.273 | 0.733 | 1.000 |
| Gemini 2.5 Flash · cross-verify | 0.000 [0,.228] | 0.000 | 1.000 | 0.919 |

Table 3: Natural (found) corpus: claims GPT-4o-mini *spontaneously emitted* on real exact questions, then gate-labeled. It hallucinates false exact claims at 23% overall (63% on fraction arithmetic; 0% on prime-naming). *Self*-verification admits a quarter of the false ones (false-accept CI excludes 0); a stronger *cross*-verifier catches them but abstains on 8% (coverage $< 1$). Only the exact gate is $(0, 0)$ at full coverage, so the trust-layer result is not an artifact of hand-crafted items.

on 8% (coverage 0.92), and itself false-accepts 27% of harder constructed claims: cross-model checking helps but is fragile in exactly the compute-hard regime that matters. Only the gate is $(0, 0)$ at *full* coverage, on found and constructed claims alike.

**A selective-prediction reading: the gate Pareto-dominates every model.** The benchmark has the standard structure of selective classification (El-Yaniv and Wiener 2010; Geifman and El-Yaniv 2017): a trust method is a classifier that may *abstain*, and its quality is a (coverage, risk) operating point, with coverage the fraction it answers and risk the error among answered. On the exact $n$=240 corpus the gate is the ideal point (coverage=1, false-accept=0, false-reject=0) and **Pareto-dominates all four frontier models**: where a model's dangerous false-accept is low, it has been paid for elsewhere. Gemini Pro reaches its hard-tier false-accept only by *abstaining on* 54% of hard items (coverage 0.46), Claude Sonnet by over-rejecting 24% of *true* claims, and GPT-5.6 Sol answers everything but at false-accept 0.075. No threshold, prompt, or abstention policy moves any of them to the corner, and even a code interpreter does not (Discussion); the gate is there by construction. The gap is a Pareto gap, not a tuning artifact (Figure 1; `results/falseclaim_selective.json`).

**The gap is an impossibility, not a tuning failure (Theorem 2 instantiated).** We measure $\mathrm{AUC}_P$ directly. Using a model's *own* 3-seed self-consistency confidence as its best internal statistic, gemini-2.5-flash separates true from

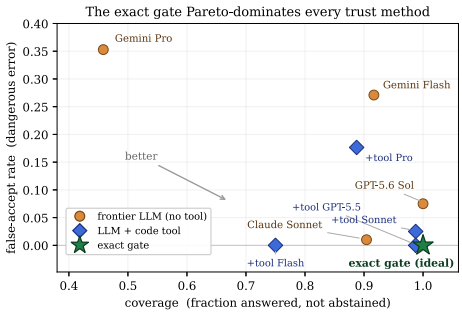



Figure 1: **Central result.** Selective-prediction view of the hard tier: false-accept (dangerous error) vs. coverage. The exact gate sits at the ideal corner (coverage=1, false-accept=0) and **strictly Pareto-dominates** every frontier model (circles) *and* every tool-augmented model (diamonds, four models). Every point sits off the corner: no-tool models by high false-accept or heavy abstention; tool-augmented models by abstention (Flash), residual false-accept where they attempt the hardest domain (Pro, 0.176), or over-rejection of true claims (Sonnet, GPT-5.5). No prompt, threshold, abstention policy, or code tool reaches the corner; only the gate is there, by construction.

| proposer | solve (blind) | false proposals | escape w/o gate | escape w/ gate |
|---|---|---|---|---|
| Gemini 2.5 Flash | 0.375 | 43 | 43 | **0** |
| Claude Sonnet | 1.000 | 1 | 1 | **0** |

Table 4: Proposer soundness. The gate makes an unreliable proposer *sound*: every returned answer is verified and every false proposal caught (0 escape), for both a 38%- and a $\sim$100%-reliable proposer.

false hard claims at only $\mathbf{AUC = 0.729}$ overall: **0.500** (chance) on ground-energy, where it is provably predicate-blind, and 0.69–0.71 on the large-number arithmetic domains, reaching 1.0 only on hadrons, the one family it can actually compute. This holds across all four frontier models ($\mathrm{AUC}_P \in [0.66, 0.87]$, *none* at the corner; appendix): weaker models miss via false-accept, stronger ones (Sonnet FPR=0, GPT-5.6-sol) via low true-accept, so it is no weak-model artifact. By Theorem 2, $\mathrm{AUC}_P < 1$ makes the corner internally unreachable *for every* confidence-, self-consistency-, or judge-based method; the exact gate is at $\mathrm{AUC} = 1$ by deciding $g$. The Pareto gap in Figure 1 is therefore information-theoretic, and the honest nuance, that an internal method *could* certify hadrons because there the model decides the predicate, is exactly what the theorem predicts (`exp_internal_certification_limit`).

## 6 Evaluation II: Verifier-Guided Generation

Does gating a proposer make it trustworthy? We task each model with Erdős–Straus decompositions (Elsholtz and Tao 2013) $4/p = 1/x + 1/y + 1/z$ (exact-checkable) over 8 primes, comparing *loop-on* (the gate's exact refutation is fed back) with *loop-off* (blind retries).

Table 4 shows that the gate makes the *system* sound regard-

| feedback design (held-out primes) | solve-rate | mean rounds |
|---|---|---|
| *deterministic proposer (isolates the feedback-design variable, 5 seeds):* | | |
| blind (full-triple guess) | 0.000 | — |
| scalar error (the negative) | 0.000 | — |
| **actionable exact sub-result** | **0.44** $\pm.02$ | 3.2 |
| **+ gate-supervised learned policy** | **0.92** $\pm.02$ | 2.1 |
| *real LLM proposers, end-to-end (scalar → staged):* | | |
| GPT-4o-mini | $0.00 \rightarrow \mathbf{0.83}$ | 1.2 |
| Gemini 2.5 Flash* | $0.00 \rightarrow \mathbf{0.92}$ | 1.6 |

Table 5: Turning the loop positive. A scalar error signal does not help (reproducing the negative); an actionable exact sub-result does ($0.00 \rightarrow 0.44$), and a gate-supervised learned $x$-policy reaches $0.92$ on held-out primes. **Two real LLM proposers confirm it end-to-end**: both solve $0/12$ held-out primes under scalar feedback but $0.83$–$0.92$ under staged feedback; *this includes the very model for which the naive loop showed no lift.

less of proposer reliability: every false proposal is caught, and none escape unverified. But does the loop *improve* the proposer? Feeding the gate's *scalar* refutation back ("your sum was off by $D$") does *not*: the solve rate is flat (Gemini $+0.000$, Sonnet $-0.125$). We first read this as a negative, and then found it is a property of the feedback *design*, not of the loop. A scalar error is not actionable. Table 5 isolates the variable with a deterministic proposer on a blind train/test split of primes, so the effect is not model noise. Replacing the scalar error with an **actionable exact sub-result** lifts the solve rate from $0.00$ to $\mathbf{0.44}$: the gate returns the exact remainder $4/p - 1/x$ and, when it exists, the completed two-term $(y, z)$, so the proposer need only iterate on $x$. A **gate-supervised learned proposal policy** (which $x$-offsets yield a completable remainder, learned on a disjoint train split) lifts it further to $\mathbf{0.92}$ on held-out primes in about two rounds. Verifier-in-the-loop therefore *does* improve the proposer, provided the verifier returns exact structure rather than a scalar and a gate-supervised prior guides the proposals. This holds end-to-end for *real* LLM proposers, not just the deterministic one (Table 5, lower block): GPT-4o-mini and Gemini 2.5 Flash, the very model for which the naive loop showed no lift, go from $0.00$ solved under scalar feedback to $0.83$ and $0.92$ under staged feedback. The gate never invents a decomposition; the entire lift is in how its exact output is used.

**Does the recipe generalize beyond Erdős–Straus?** We abstract the staged proposer into a domain protocol (propose one scalar $x$; the gate returns the exact remainder and auto-completes the rest when it can) and instantiate it on two structurally different exact problems: **Fermat factorization** ($N = a^2 - b^2$, multiplicative; propose $a$, the gate tests whether $a^2 - N$ is a square) and **sum of two squares** ($p = a^2 + b^2$ for $p \equiv 1 \pmod 4$, additive; propose $a$, the gate tests whether $p - a^2$ is a square). We run the same four arms (blind/scalar/staged/+learned) on a blind train/test split (5 seeds, 12 tries). The **actionable-feedback lift** (staged over scalar) is positive on all three domains ($+0.68$, $+0.16$, $+0.63$): the ES result is not ES-specific. The additional **learned-prior lift** is positive exactly where the winning offset transfers across instances (ES $+0.30$, Fermat $+0.38$) and neutral-to-negative for sum-of-two-squares ($-0.17$), whose representation is unique and whose offset is essentially uniform, so a learned prior has no cross-instance structure to exploit. This is an honest, mechanistic boundary rather than a blanket claim: the actionable-feedback principle is general, and the learned prior helps when, and only when, proposal offsets carry cross-instance regularity. The actionable-feedback lift also holds *end-to-end for a real LLM* on the new domain: GPT-4o-mini on held-out Fermat instances rises from $0.50$ (scalar) to $0.625$ (staged) and solves in half the rounds ($1.0$ vs $2.0$), confirming the effect is not an artifact of the deterministic proposer. As before, no gate ever invents a solution in any domain; it only certifies one.

A separate held-out test corroborates the learned-prior mechanism on a harder split: a gate-supervised next-step policy trained only on *closed* number-theory problems, tested on the Erdős–Straus *hard core* ($n$=24 primes with no known small-channel solution) under a split-*by-problem* firewall ($0$ leakage), clears the smallest-channel floor ($0.292$) by a wide margin: the policy reaches $0.583$, and a tabular in-context learner $0.708$. The prior thus transfers to genuinely unseen, hardest-known instances, a harder test than the transfer-gated lift above.

## 7 Evaluation IV: The Invariant Under Open-Domain Autonomous Self-Improvement

The benchmark (Section 5) and hardware test (Section 4) exercise the invariant where an exact gate *exists*. The sharpest remaining question is what the discipline buys where one does *not*: on open, non-decidable claims, does "the system abstains" (Section 3) leave only the untrusted proposer? We answer with the hardest composition an honesty invariant can face, an agent trying to improve *itself* from unverifiable sources, and show that the invariant converts bounded coverage from a limitation into the enabling property for safe open-ended autonomy.

**Setup: a self-improvement agent on an open domain.** We instantiate the same tiering machinery in an autonomous agent whose task has *no* exact gate: find advances in the published literature that could improve the system itself. It models the system's own improvable surfaces (the trust false-accept rate of Section 5, gate coverage, cost, held-out solve rate) as targets; it web-searches for relevant work; and for each hit it proposes a concrete incorporation. The domain is exactly the regime the invariant was designed to *not* certify: whether a paper's method will help is a judgment, not a theorem.

**Result: web-sourced self-modification cannot launder into trust.** The invariant holds under two composition modes beyond Theorem 1's original examples (tabulated in the appendix). *(i) Anti-fabrication by construction:* every proposal must carry a real source URL; an uncited claim is dropped before it is recorded, so the agent structurally cannot invent a supporting paper; a named "breakthrough"

supplied as a query is treated as a *search* and returned with a citation, or honestly not at all. *(ii) A hard tier ceiling on open evidence:* a fresh proposal is REFERENCE (a cited lead) or HEURISTIC (a suggested change), never VERIFIED; and even after a *measured* benchmark gain, the strongest evidence available without an exact law, promotion stops at EMPIRICAL, because a benchmark measurement is empirical evidence, not a machine-checkable theorem. VERIFIED remains reachable only through an exact gate. Thus the agent may propose, search, and (behind governance) act freely, yet no unverified web idea can ever contaminate the certified tier, and no measured improvement can impersonate a proof.

**The positive face: open-loop autonomy yields verified, non-trivial outputs.** The ceiling above is the safety half; a trust *floor* also makes autonomy *productive*. We run the same untrusted cheap proposer *open-loop* (no human in the loop, the gate the only filter on what is kept), but on a domain where an exact gate *does* exist: the Erdős–Straus hard core, $n{=}24$ primes with no known small-channel solution (Section 6). Soundness makes false-accept 0 by construction; the non-trivial part is that the kept set is *substantial*: a gate-supervised policy certifies $58\%$ of these hardest-known primes and a tabular learner $71\%$, disjoint from training. A concrete kept output, for the smallest hard-core prime, is the verified decomposition $4/1009 = 1/253 + 1/85100 + 1/944524900$, a result a number theorist would want, produced autonomously and certified rather than asserted. So the agent runs unattended yet returns only theorems: the open domain (no gate $\Rightarrow$ nothing VERIFIED) and the gated domain (exact gate $\Rightarrow$ a verified discovery at zero false-accept) are two faces of one property, namely that autonomy is bounded in *certification*, never in *action*.

**Why this matters.** The dominant failure mode of autonomous self-improving agents is *compounding*: an unverified "improvement" is trusted, built upon, and its error propagates. The invariant makes that failure structurally impossible without narrowing what the agent may attempt: creativity and action stay unbounded, and only *certification* is bounded. This reframes the coverage limitation of Section 3: abstention is not a dead end but the safety property that lets an open-ended agent operate honestly, exactly where LLM agents are most dangerous. The guarantee is honesty (no false certification, no laundered self-improvement), not power, consistent with the paper's stance throughout.

## 8 Discussion

**"Isn't this just verification?"** No; Theorem 2 is the answer. Certification is provably *impossible* for any proposer-internal method, so an external exact tier is not a convenience but a necessity. Prior work uses an exact checker as a *producer* or *scorer*: LeanDojo drives a prover (Yang et al. 2023), AlphaGeometry a symbolic search (Trinh et al. 2024), and verifiable-reward RL a training signal (DeepSeek-AI 2025). Ours is the orthogonal inference-time move: the checker's verdict becomes a *non-forgeable trust tier* (Theorem 1) whose necessity we prove and whose value we measure as a selective predictor that Pareto-dominates every LLM baseline (Fig. 1).

**"LLMs already do well here."** A frontier model is strong on tractable items (zero false-accept on the normal tier), but it never *certifies*: it reaches zero false-accept only by over-rejecting and fails compute-hard claims entirely. Certification is a different operation from estimation, and by Theorem 2 the gap between them is one that *scaling cannot close*.

**"Just give the model a calculator or code tool."** We tested four frontier models with a native Python code tool on the hard tier (Fig. 1; $n{=}80$). This is Theorem 2 remark (iii) in practice: a tool is a step toward an *external* decision, so it drives false-accept from $0.271$ to $\mathbf{0.000}$ exactly where a claim is a short computation (Egyptian, modular, hadron). Yet no model reaches the corner, because the model must still *decide when to invoke it*, an internal judgment. Each misses differently: Flash abstains on the whole ground-energy domain ((coverage $0.75$, false-accept $0$)), Gemini Pro attempts it and *reintroduces* the error ($0.176$), Sonnet and GPT-5.5 over-reject a quarter of *true* claims. Tool use *relocates* the failure but never removes it; a code tool is a way to *build* exact checks, not a certifier. The gate removes both the decision and the when-to-invoke judgment.

**Limitations & honest non-claims.** A reality gate exists only where an exact law does; elsewhere the system abstains, bounding coverage by design. A gate certifies its *own* exact statement, not the adequacy of a surrounding model. Our quantum-hardware sizes are classically simulable, so **no quantum advantage is claimed**; and the proposer's learned-prior lift is transfer-gated rather than universal, a boundary we characterize rather than hide. The invariant guarantees *honesty* (no false certification), not *power*.

## 9 Conclusion

Trust in AI-for-science is better engineered as a boundary than sought in the proposer. We prove certification is *impossible* for any proposer-internal method and sound only through an external exact gate, and demonstrate it negatively across a 4-domain benchmark, verifier-guided generation, nine adversarial quantum-hardware jobs, and an open-domain self-improvement agent: the gate never certified a false claim. Efficiency proposes because truth is coherence; the exact gate disposes because coherence can be checked, and it cannot be fooled. That is the deliverable.

## References


Angelopoulos, A. N.; Bates, S.; Candès, E. J.; Jordan, M. I.; and Lei, L. 2021. Learn then Test: Calibrating Predictive Algorithms to Achieve Risk Control. *arXiv preprint arXiv:2110.01052*.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*.

El-Yaniv, R.; and Wiener, Y. 2010. On the Foundations of Noise-free Selective Classification. *Journal of Machine Learning Research*, 11: 1605–1641.

Elsholtz, C.; and Tao, T. 2013. Counting the number of solutions to the Erdős–Straus equation on unit fractions. *Journal of the Australian Mathematical Society*, 94(1): 50–105.

Farquhar, S.; Kossen, J.; Kuhn, L.; and Gal, Y. 2024. Detecting Hallucinations in Large Language Models Using Semantic Entropy. *Nature*, 630: 625–630.

Geifman, Y.; and El-Yaniv, R. 2017. Selective Classification for Deep Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*. ArXiv:1705.08500.

Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.; Madotto, A.; and Fung, P. 2023. Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12): 1–38.

Kandala, A.; Temme, K.; Córcoles, A. D.; Mezzacapo, A.; Chow, J. M.; and Gambetta, J. M. 2019. Error mitigation extends the computational reach of a noisy quantum processor. *Nature*, 567: 491–495.

Kogut, J.; and Susskind, L. 1975. Hamiltonian formulation of Wilson's lattice gauge theories. *Physical Review D*, 11: 395–408.

LHCb Collaboration. 2022. Observation of an exotic narrow doubly charmed tetraquark. *Nature Physics*, 18: 751–754.

Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2023. Let's Verify Step by Step. *arXiv preprint arXiv:2305.20050*.

Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; et al. 2023. Self-Refine: Iterative Refinement with Self-Feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*. ArXiv:2303.17651.

Manakul, P.; Liusie, A.; and Gales, M. J. F. 2023. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Martinez, E. A.; Muschik, C. A.; Schindler, P.; Nigg, D.; Erhard, A.; Heyl, M.; Hauke, P.; Dalmonte, M.; Monz, T.; Zoller, P.; and Blatt, R. 2016. Real-time dynamics of lattice gauge theories with a few-qubit quantum computer. *Nature*, 534: 516–519.

Mohri, C.; and Hashimoto, T. 2024. Language Models with Conformal Factuality Guarantees. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*. ArXiv:2402.10978.

Necula, G. C. 1997. Proof-Carrying Code. In *Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*, 106–119.

Parke, S. J.; and Taylor, T. R. 1986. Amplitude for $n$-Gluon Scattering. *Physical Review Letters*, 56: 2459–2460.

Romera-Paredes, B.; Barekatain, M.; Novikov, A.; Balog, M.; Kumar, M. P.; Dupont, E.; Ruiz, F. J. R.; Ellenberg, J. S.; Wang, P.; Fawzi, O.; Kohli, P.; and Fawzi, A. 2024. Mathematical discoveries from program search with large language models. *Nature*, 625: 468–475.

Temme, K.; Bravyi, S.; and Gambetta, J. M. 2017. Error Mitigation for Short-Depth Quantum Circuits. *Physical Review Letters*, 119: 180509.

Temple, G. 1928. The theory of Rayleigh's principle as applied to continuous systems. *Proceedings of the Royal Society A*, 119: 276–293.

Trinh, T. H.; Wu, Y.; Le, Q. V.; He, H.; and Luong, T. 2024. Solving olympiad geometry without human demonstrations. *Nature*, 625: 476–482.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *International Conference on Learning Representations (ICLR)*. ArXiv:2203.11171.

Yang, K.; Swope, A.; Gu, A.; Chalamala, R.; Song, P.; Yu, S.; Godil, S.; Prenger, R.; and Anandkumar, A. 2023. LeanDojo: Theorem Proving with Retrieval-Augmented Language Models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E. P.; Zhang, H.; Gonzalez, J. E.; and Stoica, I. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*. ArXiv:2306.05685.